Status of the Claims

1. (currently amended) A method for downloading data, comprising the steps of:

establishing a connection over a communication network between a remote client and a server, said connection comprising ~~an original~~ a client connection socket;

receiving a download request from said client via said connection for download of information from said server;

constructing a copy of said ~~original~~ client connection socket;

maintaining an open connection ~~between~~ with said client via said copy of said ~~original~~ client connection socket:

closing said ~~original~~ client connection socket;

enqueueing said copy of said ~~original~~ client connection socket with copies of other sockets in a queue to await service of said request by ~~in~~ a download manager process; and

executing said download manager process to service said request by transmitting said information to said client via said copy of said ~~original~~ client connection socket.

2. (currently amended) The method according to claim 1, further comprising the step of forcing a conversion of said copy of said ~~original~~ client connection socket to a non-blocking socket.

3. (canceled).

4. (currently amended) The method according to claim 1, wherein said client comprises a plurality of clients, and said steps of establishing a connection, and constructing a copy of said ~~original~~ client connection socket, are performed substantially concurrently with respect to each of said plurality of clients.

5. (previously presented) The method according to claim 1, wherein said download request further comprises a request to download a file.

6. (original) The method according to claim 1, wherein said connection comprises a TCP session.

7. (currently amended) The method according to claim 1, wherein said ~~original~~ <u>client connection</u> socket of said connection is a blocking socket.

8. (original) The method according to claim 1, wherein said server allows a maximum number of open file descriptors, further comprising the steps of:

spawning a duplicate download manager process when said maximum number of open file descriptors is exceeded;

performing said step of receiving a download request in one of said download manager process and said duplicate download manager process; and

servicing previously pending requests in another of said download manager process and said duplicate download manager process.

9. (original) The method according to claim 1, further comprising the steps of:

accepting new connections for additional clients over said communication network;

establishing respective first threads in said server to operate said connection and said new connections;

establishing a second thread to operate said download manager process; and

communicating between said download manager process and said client and said additional clients via said connection and said new connections respectively using non-blocking I/O.

10. (currently amended) A computer software product, comprising a computer-readable medium in which computer program instructions are stored, which instructions, when read by a computer, cause the computer to perform a method for downloading files from said computer over a data network comprising the steps of:

intercepting a download request for information that is received via a first connection from a remote client, said first connection comprising ~~an original~~ a client connection socket;

constructing a copy of said ~~original~~ client connection socket;

maintaining an open connection ~~between~~ with said client via said copy of said ~~original~~ client connection socket:

installing a download manager in said computer;

enqueueing a set of data comprising said download request and said copy of said ~~original~~ client connection socket with copies of other sockets in a queue to await service of said request by said download manager; and

with said download manager servicing said request by downloading said information from said computer to said client via said copy of said ~~original~~ client connection socket.

11. (currently amended) The computer software product according to claim 10, wherein said client comprises a plurality of clients, and said steps of intercepting a download request, and transmitting a copy of said ~~original~~ client connection socket, and downloading said information are performed substantially concurrently with respect to each of said plurality of clients.

12. (currently amended) The computer software product according to claim 10, wherein said computer is further instructed to perform the steps of

after performing the step of enqueueing a set of data closing said first connection; and

forcing a conversion of said copy of said ~~original~~ client connection socket to a non-blocking socket.

13. (previously presented) The computer software product according to claim 10, wherein enqueueing said download request in said download manager, wherein said download request is a request to download a file.

14. (original) The computer software product according to claim 10, wherein said first connection comprises a TCP session.

15. (currently amended) The computer software product according to claim 10, wherein said ~~original~~ <u>client connection</u> socket of said first connection is a blocking socket.

16. (original) The computer software product according to claim 10, wherein said computer allows a maximum number of open file descriptors, further comprising the steps of:

spawning a duplicate download manager when said maximum number of open file descriptors is exceeded;

receiving said set of data in one of said download manager and said duplicate download manager; and

performing said step of download information for previously pending requests in another of said download manager and said duplicate download manager.

17. (previously presented) The computer software product according to claim 10, wherein said computer is further instructed to perform the steps of:

accepting new connections for additional clients ;

establishing respective first threads in said computer to operate said new connections;

establishing a second thread to operate said download manager; and

communicating between said download manager and said client and said additional clients via said new connections respectively using non-blocking I/O.

18. (previously presented) A system for downloading information over a data network, comprising:

a server connectable to a plurality of clients across said data network via blocking sockets, said server being adapted to intercept download requests from said clients, and to associate each of said download requests with respective copies of said blocking sockets and to maintain open connection with said clients using said respective copies; and

a download manager executing in said server that receives said download requests and said copies from said server and enqueues said download requests and said copies, said download manager being adapted to force a conversion of said copies to non-blocking sockets, said server thereupon closing said blocking sockets, wherein said download manager causes said download requests to be serviced from said server across said data network via respective ones of said non-blocking sockets.

19. (original) The system according to claim 18, wherein said download manager is a subassembly of said server.

20. (original) The system according to claim 18, wherein said download manager comprises a queue for holding said download requests, wherein said download requests are serviced in turn from said queue.

21. (previously presented) The system according to claim 18, wherein said server establishes respective first separate threads to operate sockets connecting said server with said clients and establishes a second thread to operate said download manager, said download manager force conversions of said blocking sockets to non-blocking sockets and communicating with said clients via said non-blocking sockets using non-blocking I/O.